Машиналық оқытуға кірісіспе

3- дәріс. Шешім қабылдау ағашы әдісі. Деректерді визуалдау кітапханалары жұмысы, функциялары, әдістері.

Лектор: Кабдрахова С.С.

- Говорят, что компьютерная программа *обучается* при решении какой-то задачи из класса *T*, если ее производительность, согласно метрике *P*, улучшается при накоплении опыта *E*.
- Далее в разных сценариях под *T, P*, и *E* подразумеваются совершенно разные вещи. Среди самых популярных **задач *T* в машинном обучении**:
- **классификация** – отнесение объекта к одной из категорий на основании его признаков
- **регрессия** – прогнозирование количественного признака объекта на основании прочих его признаков

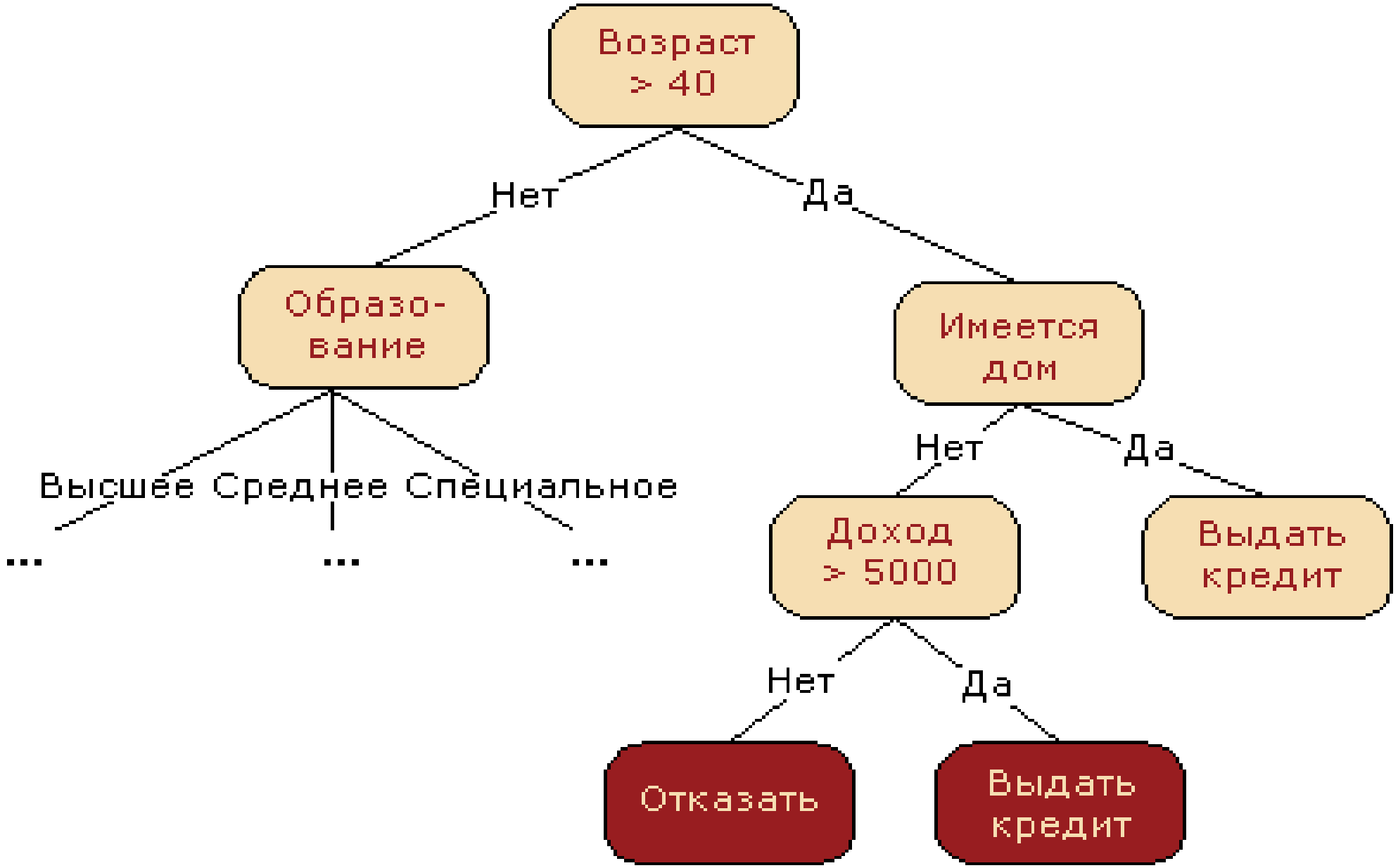
- **кластеризация** – разбиение множества объектов на группы на основании признаков этих объектов так, чтобы внутри групп объекты были похожи между собой, а вне одной группы – менее похожи
- **детекция аномалий** – поиск объектов, "сильно непохожих" на все остальные в выборке либо на какую-то группу объектов
- и много других, более специфичных. Хороший обзор дан в главе "Machine Learning basics" книги "Deep Learning" (Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016)

- Под **опытом *E*** понимаются данные (без них никуда),
- и в зависимости от этого алгоритмы машинного обучения могут быть поделены на те, что обучаются *с учителем* и *без учителя* (supervised & unsupervised learning).
- В задачах **обучения без учителя** имеется *выборка*, состоящая из *объектов*, описываемых набором *признаков*.
- В задачах **обучения с учителем** вдобавок к этому для каждого объекта некоторой выборки, называемой ***обучающей***, известен ***целевой признак*** – по сути это то, что хотелось бы прогнозировать для прочих объектов, не из обучающей выборки.
- Задачи классификации и регрессии – это задачи обучения с учителем

Классификация, деревья решений

- **Дерево решений**

- Начнем обзор методов классификации и регрессии с одного из самых популярных – с дерева решений. Деревья решений используются в повседневной жизни в самых разных областях человеческой деятельности, порой и очень далеких от машинного обучения. Деревом решений можно назвать наглядную инструкцию, что делать в какой ситуации.
- Зачастую дерево решений служит обобщением опыта экспертов, средством передачи знаний будущим сотрудникам или моделью бизнес-процесса компании. Например, до внедрения масштабируемых алгоритмов машинного обучения в банковской сфере задача кредитного скоринга решалась экспертами. Решение о выдаче кредита заемщику принималось на основе некоторых интуитивно (или по опыту) выведенных правил, которые можно представить в виде дерева решений.


Возраст
> 40
Нет
Да
Образо-
вание
Имеется
дом
Высшее
Среднее
Спёциальное
...
...
...
Нет
Да
Доход
> 5000
Выдать
кредит
Нет
Да
Отказать
Выдать
кредит

- Алгоритм дерево решений-**C4.5**, рассматривается первым в списке 10 лучших алгоритмов интеллектуального анализа данных

- **Энтропия**

- Энтропия Шеннона определяется для системы с N возможными состояниями следующим образом:

- $$S=-\mathrm{Sum}(i=1,N)\ p_i \log_2(p_i),$$

- де **pi** – вероятности нахождения системы в **i**-ом состоянии. Это очень важное понятие, используемое в физике, теории информации и других областях. Опуская предпосылки введения (комбинаторные и теоретико-информационные) этого понятия, отметим, что, интуитивно, энтропия соответствует степени хаоса в системе. Чем выше энтропия, тем менее упорядочена система и наоборот. Это поможет нам формализовать "эффективное разделение выборки", про которое мы говорили в контексте игры "20 вопросов".
-

- \Large IG(Q) = S_O - \sum_{i=1}^{q}\frac{N_i}{N}S_i,
- где q – число групп после разбиения, Ni – число элементов выборки, у которых признак Q имеет i-ое значение.

- **Алгоритм построения дерева**

- В основе популярных алгоритмов построения дерева решений, таких как ID3 и C4.5, лежит принцип жадной максимизации прироста информации – на каждом шаге выбирается тот признак, при разделении по которому прирост информации оказывается наибольшим.
- Дальше процедура повторяется рекурсивно, пока энтропия не окажется равной нулю или какой-то малой величине (если дерево не подгоняется идеально под обучающую выборку во избежание переобучения).
  В разных алгоритмах применяются разные эвристики для "ранней остановки" или "отсечения", чтобы избежать построения переобученного дерева.

- def build(L):
  - create node t
  - if the stopping criterion is True:
    - assign a predictive model to t

  else:

    Find the best binary split L = L_left + L_right

    t.left = build(L_left) t.right = build(L_right)

  return t

- Другие критерии качества разбиения в задаче классификации
- Мы разобрались в том, как понятие энтропии позволяет формализовать представление о качестве разбиения в дереве. Но это всего лишь эвристика, существуют и другие:
- **Неопределенность Джини (Gini impurity): $G=1-\sum_k(p_k)2$.** Максимизацию этого критерия можно интерпретировать как максимизацию числа пар объектов одного класса, оказавшихся в одном поддереве.
- Не путать с индексом Джини!
- **Ошибка классификации (misclassification error):** $E=1-\max\{k\} p\_k$
- На практике ошибка классификации почти не используется, а неопределенность Джини и прирост информации работают почти одинаково.

- В случае задачи бинарной классификации (p+ – вероятность объекта иметь метку +) энтропия и неопределенность Джини примут следующий вид:

$$S = -p_+ \log_2{p_+} -p_- \log_2{p_-} = -p_+ \log_2{p_+} -(1 - p_{+}) \log_2{(1 - p_{+})};$$

$$G = 1 - p_+^2 - p_-^2 = 1 - p_+^2 - (1 - p_+)^2 = 2p_+(1-p_+).$$

Когда мы построим графики этух двух функций от аргумента p+, то увидим, что график энтропии очень близок к графику удвоенной неопределенности Джини, и поэтому на практике эти два критерия "работают" почти одинаково.

- Деревья решений могут создавать сложные границы решений, разделяя пространство объектов на прямоугольники. Однако мы должны быть осторожны, так как чем глубже дерево решений, тем сложнее становится граница решения, что может легко привести к переобучению.
- Используя scikit-learn, мы теперь будем обучать дерево решений с максимальной глубиной 3, используя энтропию в качестве критерия примеси. Хотя масштабирование объектов может быть желательно для целей визуализации, обратите внимание, что масштабирование объектов не является обязательным требованием для алгоритмов дерева решений. Код выглядит следующим образом: